"Express Mail" mailing label number EV 164034163 US

Date of Deposit: December 15, 2003

Attorney Docket No. 15137US02

DMA ENGINE FOR FETCHING WORDS IN REVERSE ORDER

RELATED APPLICATIONS

[0001] This application claims priority to Provisional Application for U.S. Patent, Serial No. 60/494666, (Attorney Docket Number 15137US01) entitled "DMA Engine for Fetching Words in Reverse Order", filed August 13, 2003, and Provisional Application for U.S. Patent, Serial No. 60/494746, (Attorney Docket Number 15138US01) entitled "DMA Engine for Fetching Words in Reverse Order", filed August 13, 2003, which are incorporated herein by reference for all purposes.

[0002] This application is also related to Application for U.S. Patent, Serial No. _____, entitled "DMA Engine for Fetching Words in Reverse Bit Order", filed _____, which is incorporated herein by reference.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[0003] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[0004] [Not Applicable]

BACKGROUND OF THE INVENTION

[0004] The MPEG-2 standard uses video packets comprising any number of macroblocks to confine bit errors. Frames are represented by a set of macroblocks. The macroblocks are grouped into a data structure known as a video packet. In

MPEG-2, all macroblock rows start with a new video packet. During the decoding of a video packet, if an error is encountered, the video decoder can simply drop the remaining macroblocks that followed the bit error. In this manner, only a small amount of information is lost as a result of the bit error.

[0005] MPEG-4 Part 2 also uses video packets to confine bit errors. The video data is transmitted as a video elementary stream. The portions of the video elementary stream that are at the video packet level and lower are encoded with a variable length code. In MPEG-4 Part 2, the video packet is defined such that the video packet can be decoded in a forward order, or the reverse order. Accordingly, after encountering an error, the video decoder can go to the end of the video packet, and start decoding in the reverse order, until the same error, or another error is encountered. In this manner, a greater portion of the video packet is recovered and reconstructed, in spite of encountering error(s).

[0006] To take advantage of the foregoing feature, the video decoder needs to be able to receive and decode the video bitstream in reverse order in real-time. During the decoding, the video elementary stream is stored in a memory known as the compressed data buffer in the forward order, along with a table that indicates the starting addresses of each video packet. The video decoder receives and decodes the video elementary stream by accessing the compressed data buffer. Upon encountering an error, the video decoder can receive the video packet at the ending address of the video packet and moving in the reverse order.

[0007] Receiving the video packet in reverse order can be made possible by manipulating the memory access. For example, the video decoder can sequentially access data words in reverse order. After accessing each data word, the video decoder can use logic to reverse the bit order of the data word. However, the foregoing adds significant operations to the video decoder and makes accessing and decoding in the reverse order difficult to perform in real-time.

[0008] Further limitations and disadvantages of conventional and traditional systems will become apparent to one of skill in the art through comparison of such systems with the invention as set forth in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[0009] Presented herein is a direct memory access engine for fetching words in reverse order. In one embodiment, there is presented a method for providing a plurality of sequential data words. The method includes receiving a command to provide the plurality of sequential data words, wherein the plurality of sequential data words comprises a first data word and a last data word, and one or more data words between the first data word and the last data word, fetching a sequential portion of the sequential data words, said sequential portion comprising a first intermediate word, the last word, and one or more data words between the intermediate word and the last word, storing the sequential portion, transmitting at least a portion of the last data word, and transmitting at least a portion of the intermediate data words after transmitting at least the portion of the last data word.

[0010] In another embodiment, there is presented a system for providing a plurality of sequential data words. The system comprises a state logic machine, a memory controller, a local buffer, and a port. The state logic machine receives a command to provide the plurality of sequential of sequential data words, wherein the plurality of sequential data words comprises a first data word and a last data word, and one or more data words between the first data word and the last data word. The memory controller fetches a sequential portion of the sequential data words, said sequential portion comprising a first intermediate word, the last word, and one or more data words between the intermediate word and the last word. The local buffer stores the sequential portion. The port

transmits at least a portion of the last data word and transmits at least a portion of the intermediate data words after transmitting at least the portion of the last data word.

[0011] In another embodiment, there is presented a system for decoding a video packet. The system comprises a compressed data buffer, a video decoder, and a direct memory access engine. The compressed data buffer comprises a plurality of sequential data words. The plurality of sequential data words store a video packet. The video decoder decodes the video packet. The direct memory access engine provides the video packet to the video decoder and comprises a state logic machine, a memory controller, a local buffer, and a port. The state logic machine receives a command to provide the plurality of sequential data words and a control signal indicating reverse order from the video decoder, wherein the plurality of sequential data words comprises a first data word and a last data word, and one or more data words between the first data word and the last data word. The memory controller fetches a sequential portion of the sequential data words, said sequential portion comprising a first intermediate word, the last word, and one or more data words between the intermediate word and the last word. The local buffer stores the sequential portion. The port transmits at least a portion of the last data word and transmits at least a portion of the intermediate data words after transmitting at least the portion of the last data word.

[0012] These and other advantages and novel features of the present invention, as well as details of illustrated

embodiments thereof, will be more fully understood from he following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[0013] **FIGURE 1** is a block diagram describing the encoding of video data;

[0014] **FIGURE 2** is a block diagram describing an exemplary decoder in accordance with an embodiment of the present invention;

[0015] **FIGURE 3** is a block diagram of an exemplary compressed data buffer;

[0016] **FIGURE 4** is a block diagram of an exemplary direct memory access engine in accordance with an embodiment of the present invention; and

[0017] **FIGURE 5** is a flow diagram for accessing data words in reverse order in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0018] Referring now to **FIGURE 1**, there is illustrated a block diagram describing MPEG formatting of video data 305. The video data 305 comprises a series of frames 310. Each frame comprises two dimensional grids of luminance Y, chroma red Cr, and chroma blue Cb pixels 315. The two-dimensional grids are divided into 8x8 blocks 335, where four blocks 335 of luminance pixels Y are associated with a block 335 of chroma red Cr, and a block 335 of chroma blue Cb pixels. The four blocks of luminance pixels Y, the block of chroma red Cr, and the chroma blue Cb form a data structure known as a macroblock 337. The macroblock 337 also includes additional parameters, including motion vectors.

[0019] The macroblocks 337 representing a frame are grouped into different video packets 340. The video packet 340 includes the macroblocks 337 in the video packet 340, as well as additional parameters describing the video packet. Each of the video packets 340 forming the frame form the data portion of a picture structure 345. The picture 345 includes the video packets 340 as well as additional parameters. The pictures are then grouped together as a group of pictures 350. The group of pictures 350 also includes additional parameters. Groups of pictures 350 are then stored, forming what is known as a video elementary stream 355. The video elementary stream 355 is then packetized to form a packetized elementary sequence 360. Each packet is then associated with a transport header 365a, forming what are known as transport packets 365b.

[0020] The transport packets 365b can be multiplexed with other transport packets 365b carrying other content, such as another video elementary stream 355 or an audio elementary stream. The multiplexed transport packets from what is known as a transport stream. The transport stream is transmitted over a communication medium for decoding and presentation.

[0021] Referring now to **FIGURE 2**, there is illustrated a block diagram of an exemplary decoder for decoding compressed video data, configured in accordance with an embodiment of the present invention. A processor, that may include a CPU 490, reads a stream of transport packets 365b (a transport stream) into a transport stream buffer 432 within an SDRAM 430.

[0022] The data is output from the transport stream presentation buffer 432 and is then passed to a data transport processor 435. The data transport processor then demultiplexes the MPEG transport stream into its PES constituents and passes the audio transport stream to an audio decoder 460 and the video transport stream to a video transport processor 440.

[0023] The video transport processor 440 converts the video transport stream into a video elementary stream and provides the video elementary stream to an MPEG video decoder 445 that decodes the video. The video elementary stream 355 is stored in a compressed data buffer (CDB) 447. The MPEG video decoder 445 accesses the compressed data buffer (CDB) to receive the video elementary stream 355. The video elementary stream 355 is decoded by the MPEG video decoder 445 resulting in the reconstructed video data 305.

[0024] The audio data is sent to the output blocks and the video data 305 is sent to a display engine 450. The display engine 450 is responsible for and operable to scale the video picture, render the graphics, and construct the complete display among other functions. Once the display is ready to be presented, it is passed to a video encoder 455 where it is converted to analog video using an internal digital to analog converter (DAC). The digital audio is converted to analog in the audio digital to analog converter (DAC) 465.

[0025] Referring now to **FIGURE 3**, there is illustrated a block diagram describing an exemplary compressed data buffer 447. The compressed data buffer 447 includes any number of data words 505(1)...505(m). The data words can have any width. In an exemplary case, for example, the data words can comprise 256 bit jumbo words (words).

[0026] The compressed data buffer 447 stores the video elementary stream 355. The video elementary stream 355 comprises any number of video packets 340. The video packets 340 further comprise a video packet header and any number of macroblocks 337. The compressed data buffer 447 also stores a start code table 507. The start code table 507 associates each video packet 340 with its starting address in the compressed data buffer 447. Alternatively the video packet 340 can be first scanned forward, without evacuating data from memory, then number of bytes/bits can be counted, and then returned using the DMA engine.

[0027] The MPEG video decoder 445 receives the video packets 340 from the video elementary stream 355 and decodes the video packets 340. The video packet 340 is received and decoded by the MPEG video decoder 445 starting

from the word 505(x) storing the beginning of the video packet 340, and proceeding to the word 505(n) storing the end of the video packet 340.

[0028] A direct memory access (DMA) engine 510 facilitates receipt of the video packets 340 by the MPEG video decoder 445. Alternatively, a processor can facilitate receipt of the video packets 340. Accordingly, DMA engine 510 shall be interpreted to also include a processor that is operable to fetch video packets from memory. The MPEG video decoder 445 receives a video packet 340 by looking up the starting address and the ending address of a video packet 340 in the start code table 507. The MPEG video decoder 445 can then command the DMA engine 510 to fetch the words 505(x)...505(n) that store the video packet 340. Responsive thereto, the DMA engine 510 fetches and provides the words 505(x)...505(n) that store the video packet 340.

[0029] The DMA engine 510 provides the words 505(x)...505(n) to an extractor 515 within the MPEG video decoder 445 in a serial manner, beginning with word 505(x) and proceeding to the last word 505(n). The MPEG decoder 445 decodes the video packet 340, in a serial manner, beginning decoding with the first word 505(x) and proceeding to the last word 505(n). The extractor 515 and the DMA engine 510 operate in conjunction with each other, such that the words 505 are provided to the MPEG video decoder 445 at a dynamic rate that is in substantial relationship to the rate that the MPEG video decoder 445 is decoding the words 505.

[0030] In MPEG-4 PART 2, the video packet 340 is defined such that the video packet 340 can be decoded in a forward

order, or the reverse order. Accordingly, if the MPEG video decoder 445 encounters an error, the video decoder can go to the end of the video packet 340, and start decoding in the reverse order. For example, if the MPEG video decoder 445 decodes the video packet 340 beginning with the first word 505(x), and encounters an error in word 505(x+5), the MPEG video decoder 445 can start decoding the video packet 340 from word 505(n) and decode in the reverse order, e.g., word 505(n-1), 505(n-2)..., etc.

[0031] Upon detecting an error, the MPEG video decoder 445 transmits a command to the DMA engine 510 to fetch the words storing the video packet 340, e.g., words 510(x)...510(n), along with a reverse order signal. Responsive thereto, the DMA engine 510 provides the words 510(x)...510(n) in the reverse order to the MPEG video decoder 445.

[0032] Referring now to **FIGURE 4**, there is illustrated a block diagram describing an exemplary DMA engine 510 in accordance with an embodiment of the present invention. The DMA engine 510 comprises a state logic machine 605, a local buffer 610, and a memory controller 620. The local buffer 610 can comprise any amount of memory with any width of data words. For example, in an exemplary case, the memory can comprise 128 32-bit words 611(0)...611(127).

[0033] The state logic machine 605 receives a command to fetch data words in an address range, e.g., 505(x)-510(n) from the MPEG video decoder 445. The command can be accompanied by a control signal indicating that the data words in the address range are to be provided to the MPEG video decoder 445 in the reverse order, e.g., 505(n), 505(n-1)...505(x).

[0034] Responsive to receiving a command to fetch the data words 510(x)-510(n) in the reverse order, the state logic machine 605 commands the memory controller 620 to retrieve a batch comprising a the last predetermined number of data words 505 in the provided address range, and store the predetermined number of data words in the local buffer 610. The predetermined number of data words in the batch is less than or equal to the capacity of the local buffer 610. For example, in an exemplary embodiment, where the local buffer 610 comprises 128 32-bit words, the batch of data words 505 can include the last 16 words in the provided address range, e.g., data words 505(n-15)...505(n).

[0035] After the batch of data words 505(n-15)...505(n) is stored in the local buffer 610, the state logic machine 605 causes the contents of the local buffer 610 to be provided to the MPEG video decoder 445 beginning with word 611(127), and proceeding sequentially to word 611(0). After the contents of the local buffer 610, e.g., words 611(127)...611(0), are provided to the MPEG video decoder 445, the state logic machine 605 commands the memory controller 620 to fetch another batch comprising the predetermined number of words, e.g., data words 505(n-31)...505(n-16), that precede the most recently fetched data words, e.g., data words 505(n-15)...505(n). The data words 505(n-15)...505(n) are stored in the local buffer 610 and provided to the MPEG decoder 445.

[0036] The foregoing is repeated until the next predetermined number of data words comprises the first data word in the address range, e.g., data word 505(x). Where a batch comprises the first data word in the address range, e.g., data word 505(x), the state logic machine 605

truncates that portion of the predetermined number of data words that precedes the first data word 505(x), and commands the memory controller 620 to fetch the truncated batch comprising the first data word 505(x) and all data words 505(x+1), 505(x+2)..., following the first data word 505(x) that have not been previously transmitted to the MPEG video decoder 445.

[0037] The foregoing provides the data words 505(x)...505(n) as a set of 32 bit words starting from the last portion of 505(n) and proceeding sequentially to the first portion of 505(x). The bits forming the 32-bit words can be reversed with respect to one another, in any number of ways. For example, the MPEG video decoder 445 can include logic that reverses the 32 bits of each word. Alternatively, the DMA engine 510 can include additional circuitry that causes the 32 bits of each word 611 to be provided to the MPEG video decoder 445 in the reverse order.

[0038] Referring now to **FIGURE 5**, there is illustrated a flow diagram for providing a video packet in a reverse order. At 705, the state logic machine 605 receives a command to fetch data words in an address range, e.g., 505(x)-510(n), from the MPEG video decoder 445, accompanied by a control signal indicating that the data words in the address range are to be provided to the MPEG video decoder 445 in the reverse order, e.g., 505(n), 505(n-1)...505(x).

[0039] Responsive to receiving the command, the state logic machine 605 determines (706) if a predetermined number of words comprises the first word, 505(x). Where the predetermined number of words comprises the first data word in the address range, e.g., data word 505(x), the state

logic machine 605 truncates that portion of the predetermined number of data words that precedes the first data word 505(x) and commands (708) the memory controller 620 to fetch (709) the truncated batch comprising the first data word 505(x) and all data words 505(x+1), 505(x+2)..., following the first data word 505(x) that have not been previously transmitted to the MPEG video decoder 445. Where during 706, the state logic machine 605 determines that the predetermined number of words does not comprise the first word, 505(x), the state logic machine 605 commands (710) the memory controller 620 to fetch (715) a batch comprising the last predetermined number of data words 505 in the provided address range.

[0040] The fetched data words 505 are stored (720) in the local buffer 610. After the data words are stored in the local buffer 610, the state logic machine 605 causes the contents of the local buffer 610 to be provided (725) to the MPEG video decoder 445 beginning with word 611(127), and proceeding sequentially to word 611(0). After the contents of the local buffer 610, e.g., words 611(127)...611(0), are provided to the MPEG video decoder 445, a determination (730) is made whether the first data word, data word 505(x) has been provided to the MPEG video decoder 445. If the first data word has not been provided to the MPEG video decoder at 730, 705-730 are repeated. If the first data word has been provided to the MPEG video decoder at 730, the process is completed.

[0041] One embodiment of the present invention may be implemented as a board level product, as a single chip, application specific integrated circuit (ASIC), or with varying levels integrated on a single chip with other

portions of the system as separate components. The degree of integration of the system will primarily be determined by speed and cost considerations. Because of the sophisticated nature of modern processors, it is possible to utilize a commercially available processor, which may be implemented external to an ASIC implementation of the present system. Alternatively, if the processor is available as an ASIC core or logic block, then the commercially available processor can be implemented as part of an ASIC device with various functions implemented as firmware.

[0042] While the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment(s) disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.